

What is claimed is:

1. A method for collecting and displaying object interaction on a target comprising the steps of:

logging object interaction data on a target over a monitoring period;

displaying the object interaction data as a graph, the graph having a plurality of nodes and at least one line, each node being associated with a corresponding object, each line connecting two of the nodes and representing an interaction between the respective objects associated with the two nodes.

2. The method of claim 1, wherein the displaying step comprises displaying the graph on a host computing environment, and wherein the method further comprises the step of, prior to the displaying step, uploading the logged object interaction data from the target to the host computing environment.

3. The method of claim 1, wherein the object interaction data is operating system object interaction data.

4. The method of claim 1, wherein the step of logging object interaction data comprises recording interactions between operating system objects, the operating system objects including one or more of a semaphore, an ISR, a task and a memory call.

5. The method of claim 1, wherein each line is terminated with at least one arrow.

6. The method of claim 5, wherein each arrow on each line is indicative of a direction of interaction between the two nodes connected by said each line.

7. The method of claim 1, further comprising the step of accepting, as input from a user, a query for information regarding the logged object interaction data, and displaying information responsive to the query to the user.

8. The method of claim 1, wherein the logging step is performed by a WindView development tool.
9. The method of claim 8, wherein the WindView development tool is a WindView 2.0 development tool.
10. The method of claim 9, wherein the WindView development tool is a WindView 1.0 development tool.
11. The method of claim 1, wherein the logging step is performed by a Linux Trace Toolkit development tool.
12. A system comprising:
 - a target environment;
 - a display;
 - an executable logging component, the executable logging component logging object interaction data on the target environment over a monitoring period;
 - an executable graphing component, the executable graphing component displaying the object interaction data as a graph on the display, the graph having a plurality of nodes and at least one line, each node being associated with a corresponding object, each line connecting two of the nodes and representing an interaction between the respective objects associated with the two nodes.
13. The system of claim 12, wherein the executable logging component executes in the target environment.
14. The system of claim 12, wherein the executable graphing component and the executable logging component both execute on a common processor.
15. The system of claim 12, further comprising a host computing environment coupled

to the target environment and to the display, the executable graphing component executing on the host computing environment.

16. The system of claim 15, wherein the executable logging component executes on the target environment.

17. The system of claim 16, wherein host computing environment uploads the logged object interaction data from the target environment.

18. The system of claim 12, wherein the object interaction data is operating system object interaction data.

19. The system of claim 12, wherein the executable logging component records interactions between operating system objects, the operating system objects including one or more of a semaphore, an ISR, a task and a memory call.

20. The system of claim 12, wherein each line is terminated with at least one arrow.

21. The system of claim 20, wherein each arrow on each line is indicative of a direction of interaction between the two nodes connected by said each line.

22. The system of claim 12, wherein the executable graphing component accepts, as input from a user, a query for information regarding the logged object interaction data, and displays information responsive to the query to the user.

23. The system of claim 12, wherein the executable logging component is a component of a WindView development tool.

24. The system of claim 23, wherein the WindView development tool is a WindView 2.0 development tool.

25. The system of claim 23, wherein the WindView development tool is a WindView 1.0 development tool.
26. The system of claim 12, wherein the executable logging component is a component of a Linux Trace Toolkit development tool.
27. A host computing environment for accepting data from a target processor, comprising:
an executable component for executing on a host computing environment, the component receiving, from a target environment, object interaction data for objects executing in the target environment, the executable component displaying the object interaction data as a graph on the display, the graph having a plurality of nodes and at least one line, each node being associated with a corresponding object, each line connecting two of the nodes and representing an interaction between the respective objects associated with the two nodes.
28. The host computing environment of claim 27, wherein each line is terminated with at least one arrow.
29. The host computing environment of claim 28, wherein each arrow on each line is indicative of a direction of interaction between the two nodes connected by said each line.
30. The system of claim 27, wherein the executable component accepts, as input from a user, a query for information regarding the logged object interaction data, and displays information responsive to the query to the user.
31. A computer readable media, having stored thereon, computer executable process steps operable to control a computer to display object interaction data as a graph on a display, the graph having a plurality of nodes and at least one line, each node

being associated with a corresponding object, each line connecting two of the nodes and representing an interaction between the respective objects associated with the two nodes.

32. The method of claim 1, further comprising the steps of:

identifying isolated objects based upon the logged interaction data; and
generating a protection domain for the isolated objects.

33. The method of claim 32, wherein the protection domain is a VxWorks AE protection domain.